Surveillance Coverage and Vulnerability Awareness Concepts for Tactical Swarms

Anne-Laure Jousselme
Defence R&D Canada – Valcartier, 2459 Pie XI North, Quebec, QC Canada, G3J 1X5
Anne-Laure.Jousselme@drdc-rddc.gc.ca

Dominic Larkin
Department of Electrical Engineering and Computer Science, US Military Academy, West Point, NY 10996
Dominic.Larkin@usma.edu

Kevin Huggins
Department of Electrical Engineering and Computer Science, US Military Academy, West Point, NY 10996
Kevin.Huggins@usma.edu

Nicolas Léchevin
Defence R&D Canada – Valcartier, 2459 Pie XI North, Quebec, QC Canada, G3J 1X5
Nicolas.Lechevin@drdc-rddc.gc.ca

Patrick Maupin
Defence R&D Canada – Valcartier, 2459 Pie XI North, Quebec, QC Canada, G3J 1X5
Patrick.Maupin@drdc-rddc.gc.ca

ABSTRACT

Currently light infantry soldiers do not have access to many of their cyber resources the moment they depart the forward operating base (FOB). Commanders with recent combat experience have reported on the dearth of computing abilities once a mission is underway [17]. To address this, our group seeks to develop a tactical, mobile cloud implemented on a swarm of semi-autonomous robots. In this paper, we propose a pattern recognition approach to network vulnerability assessment applied to a tactical swarm of robots to enhance their strategy for surveillance coverage. Our work enhances network-enabled persistent surveillance within a dynamic, mobile domain via the implementation of sensor awareness concepts.

1.0 INTRODUCTION

The aim of this paper is to present preliminary results obtained with a novel methodology aimed at improving the motion and surveillance strategies of a mobile tactical swarm of robots. The problem of detecting, and maintaining target identification in realistic battlefield conditions is among the most difficult task facing the military today. For autonomous systems performing surveillance tasks, such as the tactical swarms described in this work, one of the major threats lies in the swarm's individual robot's lack of self-awareness. In this work we limit our definition self-awareness to inter-robot connectivity.

The objective for the swarm then is to perform surveillance in as economical manner as possible. This implies successfully combining the swarm's goal of dispersing itself over a wide area while reducing unnecessary coverage duplicity at the local node level. Depending on factors such as the local sparseness of the surveillance coverage, terrain complexity and other environmental impediments, the individual robots are at risk of losing contact with the rest of the swarm while performing their surveillance tasks. Furthermore, a robot may be lost accidentally, a lost caused for example by a mechanical failure or enemy action. In some situations, a lost can be benign to the swarm, *i.e* losing a robot at the external boundary of the swarm. In other

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2013**	2. REPORT TYPE	3. DATES COVERED **00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE **Surveillance Coverage and Vulnerability Awareness Concepts for Tactical Swarms**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of Electrical Engineering and Computer Science,US Military Academy,West Point,NY,10996**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**13**	

circumstances the lost can be catastrophic if the lost of a robot breaks the swarm into subcomponents.

The general problem addressed in this paper can be casted as a Situation Analysis (SA) problem. The aim of SA in a decision-making process is to provide and maintain a state of situation awareness for an agent observing a scene. For the purpose of the presented research, situation awareness also includes self-awareness. A critical function in the SA process is the real-time recognition of events and situations. More precisely, Situation Recognition is the action of identifying a situation to be something previously known. In the context of the present surveillance swarm monitoring problem, this entails swarm members automatically identifying situations that could pose a connectivity threat to the swarm.

The proposed methodology formulates the Situation Recognition problem as a typical pattern recognition problem. The solution then is a classifier designed and trained on a set of features extracted from the swarm network. The labels associated with the set of extracted network features are obtained from extensive simulations and identify vulnerable positions for robots within the swarm under various conditions. The resulting learned configurations are then used to obtain more robust motion and surveillance strategies.

2.0 TACTICAL SWARM FOR SURVEILLANCE AND COMMUNICATION

The authors consider a sensing coverage problem by a swarm of robots performing basic surveillance tasks for which each robot is responsible for maintaining acceptable global surveillance coverage for a given area even during network disturbances caused by the robot's detection tasks. Hence, this robot swarm coverage problem is two-fold: one has to (1) maintain surveillance on a detected target of interest while (2) maintaining the swarm's global coverage. Thus, the robot swarm must eventually be equipped with a strategy for recovering back to an equilibrium state in reaction to a perturbation. In addition, one will also require that both the original network structure and the robots' motion strategies are robust enough to absorb small perturbations.

2.1 Problem formalization

First, we consider the elements of our domain. Let $R=\{r_1, \ldots, r_N\}$ be the set of robots and $C=\{c_1, \ldots, c_M\}$ the set of clients. The set C combined with their spatial location is a *configuration*. We denote ρ as a robot's unique communication range, which could be adjusted based on environmental demands. Next, $E=\{e_1, \ldots, e_N\}$ is the set of communication links between the robots and $G_c=(R, E)$ is the corresponding communication graph. Let N_c, be the set of node coordinates. Two robots r_i and r_j are separated by a distance of d_{ij} and are connected if their distance is less than ρ. In later sections of this work, d_{ij} will denote the distance between two robots, two clients, or one robot and one client. We assume that (*i*) indirect links are possible through intermediate nodes acting as relays, and (*ii*) at least one of the nodes is connected to an external communication node such as a satellite or UAV. In other words, we assume that there exists a communication resource capability within the network to ensure that clients' messages are handled properly through the mobile cloud via an external wide range communication relay.

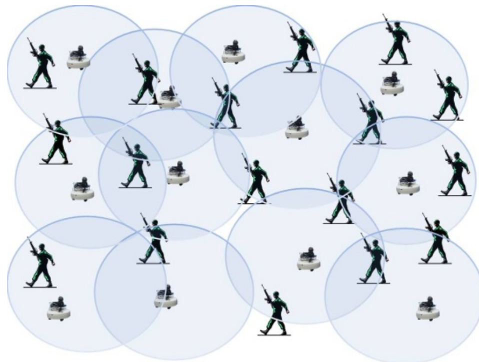

Figure 1: A tactical mobile cloud for communication coverage. The robot nodes r_i are surrounded by light blue circle that represent their coverage. ρ is the communication range and d_{ij} is the distance between robot r_i and r_j.

The strategy for the swarm must thus meet the objectives of (1) maintaining the global network connections (*i.e.*, avoid loss of connectivity), (2) ensuring the clients' coverage (*i.e.*, no client is unconnected) and (3) maintaining a global sensing coverage (*i.e.*, every intrusion within the network will be detected).
The overall objective of the swarm of robots is thus threefold:

1. Maintain the network's connectivity;
2. Maintain the clients' communication coverage;
3. Maintain an acceptable level of sensing coverage.

These three objectives can be expressed in term of coverage as it will be detailed in Section 3.0. The initial state is an equilibrium state in which the three objectives above are satisfied. Given the stochastic nature of the clients' move as well as of the robots' performances, this equilibrium state may be weakened, and two major causes of such a weakness are:

a) A loss of a node (robot's failure);

b) A loss of a link (caused by a client's move, or an obstacle to the communication link).

We assume that the swarm of robots has a motion strategy to recover that state of equilibrium and ensure the clients' coverage, the network connectivity and the network sensing coverage. We assume that both the original network structure and the motion strategy are robust enough to absorb small perturbations. But what if these perturbations grow larger?

In order to cope with this possible issue, we propose a methodology for network vulnerability assessment used as an early warning mechanism that will allow each network elements (nodes or links) to evaluate the consequences its own loss. The result of this assessment would then be used to modify the motion strategy.

2.2 Motion strategies

Each client knows its own location via a location finding device such as a GPS. We model the continuous behavior of clients and robots as a sequence of instances. At each time instance, each robot evaluates the current position of its clients as well as the set of neighboring robots. Based on this evaluation, each robot calculates the optimum position to provide coverage to clients as well as to maintain connectivity to at least one neighboring robot. If both goals cannot be attained, then the priority is to maintain coverage for clients.

At the initial equilibrium state, the network optimally covers the set of clients (communication coverage), optimally covers the area under consideration (sensing coverage) and is fully connected (each client is able to communication with an external node relaying the communication). That means in particular that (1) each client is within the communication range of at least one robot (clients' coverage) and (2) each robot is within the communication range of at least one other robot (network's connectivity).

Our robot motion strategy is based on the work in [2]. Each robot in the swarm moves according to spring-mass virtual physics. In particular, the motion of the i^{th} node in a swarm of N robots is as follows:

$$\ddot{X}_i = \left[\sum j \in S_i \, k_{ij} \left(l_{ij} - l_{ij}^0 \right) \hat{d}_{ij} \right] - \gamma_i \dot{X}_i \tag{1}$$

with i=1,..,N_r and $i \neq j$.

$\dot{X}$ and $\ddot{X}$ are the robot's velocity and acceleration respectfully, both based on the robot's position vector X. S_i is the set of robot neighbors for the i^{th} robot while l_{ij} is the length of the virtual spring between i^{th} and j^{th} robot. The symbol l_{ij}^0 represents the spring's relaxed length while $\hat{d}_{ij}$ is the unit vector indicating the direction of the spring force. Finally, the equation uses two constants: k_{ij} and γ_i. The former is the spring constant between robots i and j and the latter the damping coefficient with a value assumed to be greater than zero.

The spring-mass model described above could create a mesh with limited expandability and thus restrict a swarm's ability to cover a region adequately. To overcome this challenge, we have implemented the inter-node spring-mass links within the constraints of a Gabriel graph.

In particular, a spring is formed between two robots i and j if and only if there is no k robot inside the circle with a diameter formed by $\overline{ij}$ [2]. More formally, consider φ that evaluates to 1 if there is a spring between robots i and j and 0 otherwise. Hence, we have the following:

$$\varphi_{ij} = \begin{cases} 1 \; if \; \widehat{xkj} \leq \pi/2 \\ 1 \; if \; \widehat{xkj} > \pi/2 \end{cases} \tag{2}$$

with i, j, k=1,..,N_r and $i \neq j$ and $j \neq k$.

where $\widehat{xkj}$ is the interior angle formed by robots x, k and j.

Given the supporting equations above, we provide the following distributed algorithm that each robot in the swarm follows.

```
for each robot_j neighbor of robot_i do
    while l_ij ≠ l0_ij do
          Compute the next move from (1) subject to the constraints in (2)
          if robot_i detects a user then
                place a spring connection between robot_i and the user using (2)
          end if
          return Ẍ, Ẋ and X
    end while
end for
```

3.0 SWARM COVERAGE OBJECTIVES

Three objectives to be maintained by the swarm of robots rely on coverage notions. Indeed, the problem addressed is in essence a coverage problem involving two types of coverage:

(1) *sensing coverage*, since the swarm is responsible for covering a given area and detect possible intruders, and
(2) *communication coverage*, that can be split into:
   a. *clients' coverage*, since the swarm of robots is responsible for providing communication coverage to the set of clients within the given area, and
   b. *robots' coverage*, since the swarm is responsible for maintaining the network connectivity.

We consider the following general definition of coverage which encompasses the three notions above. Let q be an emitter p be a point of interest of a given area. The coverage provided by q at p is:

$$cov(p,q) = f(d_{pq}) \tag{3}$$

where d_{pq} is the distance separating p from q and f is a decreasing function. Both p and q are located in space and are represented by their spatial coordinates along x and y axes, *e.g.* (x_p,y_p).

3.1 Connectivity

As an instance of Equation (3), the network connectivity is defined as follows. We say that two robots are connected if they are in their respective range of communication. We define then:

$$\begin{aligned} con(r_i,r_j) &= 1 \quad \text{if } d_{ij} < \rho_c \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{4}$$

where ρ_c is the communication range of the robots and d_{ij} is the distance separating r_i form r_j. f is thus a step function of the distance and a value of 1 means then that a link exists between the two robots. We say then that the global network connectivity holds if for each pair of robots (r_i,r_j) there exists a path linking r_i to r_j:

$$con(R) = 1 \text{ if } \forall (r_i,r_j) \in R^2, \exists (r_1,\ldots,r_m) \in R^m; (r_i,r_1),(r_1,r_2)\ldots,(r_m,r_j) \in E \tag{5}$$

Alternative definitions could easily replace these binary definitions and define different connectedness indices such as the algebraic connectivity [8]. Indeed, allowing real values for the connectivity would increase our flexibility in the definition of vulnerable states and lead to different cost functions (see Section 4.0).

3.2 Clients' communication coverage

For simplicity, we use a binary definition of coverage: covered or not covered. However, this definition could be easily extended to other models, such as probabilistic ones.
We define coverage provided by robot r_i to client c_j as

$$\begin{aligned} cov(r_i,c_j) &= 1 \quad \text{if } d_{ij} < \rho_c \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{6}$$

where ρ is the communications range of r_i and d_{ij} is the distance between r_i and a client. The set of clients covered by robot r_i is then given by:

$$cov(r_i,C) = \sum_{j=1}^{M} cov(r_i,c_j) \tag{7}$$

Given the coverage definitions from the perspectives of both the client and the robot, Equation (8) describes the *global coverage* of the network relative to the set of clients.

$$cov(R,C) = \sum_{i=1}^{N} cov(r_i, C) = \sum_{j=1}^{M} cov(R, c_j) = \sum_{i=1}^{N} \sum_{j=1}^{M} cov(r_i, c_j) \tag{8}$$

3.3 Sensing coverage

We use a more precise model for defining the sensing coverage as a distance degradation model. Hence, the sensing coverage provided by a robot r_i at a given point of interest p is given by the following exponential model:

$$cov(r_i, p) = \begin{cases} 1 & \text{if } d < R \\ \exp(-\lambda(d_{ip} - R)) & \text{else} \end{cases} \tag{9}$$

where λ and β are real parameters, d_{ip} is the distance between robot r_i and the point of interest p, R is the sensor range. Figure 2 is an example of this degradation model that will be used in the simulations of Section 5.0. Within a range of 16 meters, the coverage (*i.e.*, probability of detection) is assumed to be perfect while decreasing to reach 0 around 30 meters.

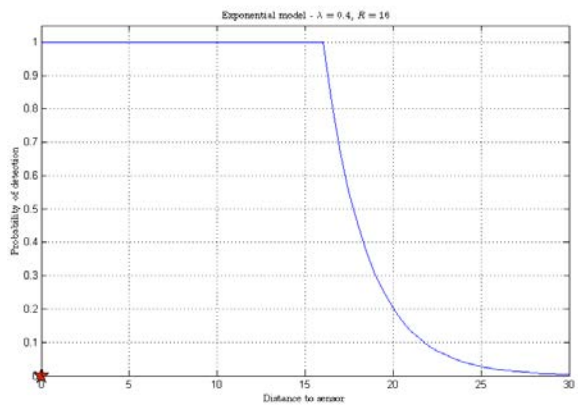

Figure 2: Exponential model of detection performance degradation according to the distance between robot r_i (represented here as a red star).

Different notions of detection (or sensing) coverage can then be computed for each node, each position of interest or the whole network. Figure 3 shows a network of 100 robots and 572 positions of interest. Each circle represents the coverage at 16 meters according to the exponential model described above.

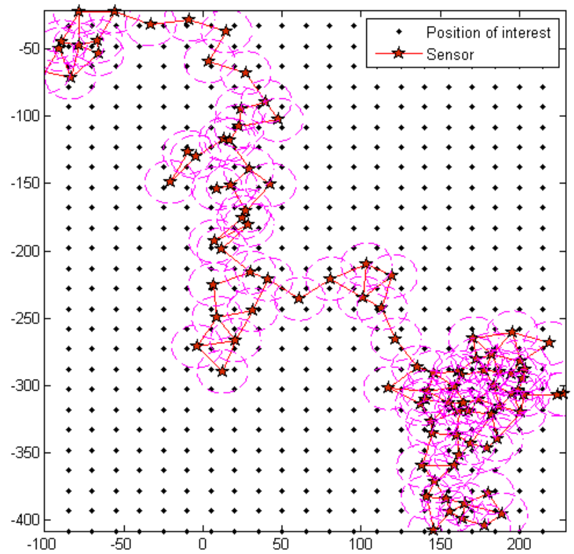

Figure 3: A tactical mobile cloud of 100 robots for sensing coverage of 572 positions of interest. Robots are represented by red stars, while positions of interest are black dots. The pink circles represent the sensing coverage of each node at 16 meters.

4.0 VULNERABILITY SELF-AWARENESS

4.1 Network vulnerability

The vulnerability V of a system S can be understood as a mapping, V_S: $T \mapsto \mathcal{C}$, between an initiating threat T, whether intended or not, and a resulting consequence $\mathcal{C}$ characterized by a degree of loss [9] and related to system inoperability or state unreachability. Depending on how the threat uncertainty is characterized, the cost function may be aggregated, giving rise to an expected cost, or equivalently to a risk function [1]. Vulnerability thus corresponds to the susceptibility of a system or to the manifestation of the inherent state of a system, which can be severely affected when threatened [9].

In this paper, a mobile network *vulnerable* state can be defined as an instance of the network's state that may evolve in time until it affects the network's functions and the completion of its goals. Endogenous and exogenous threats to the network include the robots' inability to proceed as intended, possibly due to hardware-software failures or malevolent acts, electronic warfare, obstacles, or unexpected client moves that cause some robots to move beyond their neighbors' communication range. A component of the network (*i.e.*, edge, node, or sub-network) is classified as *vulnerable* when a graph connectedness-related cost associated to this component is above a prescribed threshold. A node is thus *vulnerable* if its loss (failure) leads to a break in the network connectivity.

The idea to be detailed in the next section is to learn from a training dataset which nodes are susceptible to cause a loss of the network connectivity, depending on their structural features.

4.2 Vulnerability assessment

4.2.1 A pattern recognition approach

These principles for network vulnerability detection described in [15] rely on pattern recognition techniques that leverage structural, dynamical, and functional features selected to sensitize the classifier to potential vulnerabilities in abnormal situations. Such an approach is expected to yield fast vulnerability prediction when

compared with a simulation using a first-principle-based model of the network. To determine vulnerability, we reason over the network using pattern recognition as we proposed in [15]. With it, we design by training a mapping ψ such that:

$$\begin{gathered} \psi : G \rightarrow \{y; \bar{y}\} \\ x \mapsto \psi(x) = \hat{y} \end{gathered} \tag{10}$$

where x is a representation of an element of G (*e.g.* a node, link or sub-graph) and $\hat{y}$ is an estimate of the detrimental effect of that element on the network, either 1 if vulnerable or 0 otherwise. Typically, x is a vector of k network features identified as relevant by feature selection pre-processing. As discussed in Section 4.3, one of the crucial tasks consists in identifying the set of candidate features for the problem.

4.2.2 Assigning labels for training

Consider a sample set of possible clients configurations C_0 and a corresponding robot deployment represented by graph $G_0=(R, E_0)$. Include also the set $N_{c,0}$ of nodes' coordinates at time instant t_0 (encoded as attributes of the nodes). Various experiments are conducted by triggering the loss of a robot. The occurrence at t_1 of this triggering event gives rise to an adaptive robot deployment (see the strategy in Section 2.2), whereby communication links can be either permanently lost or re-established, depending on the relative distance to neighboring robots.

This hybrid dynamical system is characterized by switching time instants $\{t_1, t_2, \ldots, t_m\}$, where $t_{i+1}>t_i$. At t_i, the edge set jumps from E_i to E_{i+1}. An edge (i,j) is lost whenever the distance between robots r_i and r_j is greater than the communication range. It is assumed that the node set R remains invariant whether or not a robot is able to operate. The final time instant t_m is defined by the absence of any future triggering events such as a robot failure or a client move.

4.2.3 Feature extraction

Among the four network feature categories that have been proposed in [15] (*structural*, *dynamic*, *complex* and *functional*), we consider here mainly the structural features plus some functional features. Structural features pertain to the structural properties of a labeled, weighted graph and include centrality, similarity connectivity, shortest path metrics, clustering coefficient, spectral properties, vertex coreness, graph density, average nearest neighbor degree, among others [4].

The group of functional features represent functional information on key components of the network and in our application includes the notions of coverage introduced in Section 3.0.

$$\mathbf{z}^r_{m+1} = \begin{bmatrix} x^1_1 & x^1_2 & \dots & x^1_m \\ x^2_1 & x^2_2 & \dots & x^2_m \\ \vdots & \vdots & \vdots & \vdots \\ x^r_1 & x^r_2 & \dots & x^r_m \end{bmatrix} \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^r \end{bmatrix}$$

Number of clients covered	
Vertex betweenness	
Degree	
Closeness	
Alpha centrality	
Eigenvector centrality	
Page rank	
Average nearest neighbor degree	
Graph strength	
Transitivity	
Kleinberg's authority score	
Graph coreness	

Table 1.1: Training dataset with *m* features and *r* samples. x_j^i is the feature *j* extracted from sample *i*. y^i is the label (ground truth) of sample *i*. A sample is a node, and its possible labels are Vulnerable or Non-vulnerable.	**Table 1.2: List of features extracted from each node and used to estimate its vulnerability.**

Table 1: Features and corresponding labels obtained from an experiment.

4.3 Awareness

The notion of awareness considered in this work is derived from the concept resource-bounded awareness first presented in [10] and developed in [11]. Intuitively, awareness is an epistemic state, close to knowledge, referring to a limited view and a limited capacity of the agents to reach a perfect state of knowledge, the one that would be reached by perfect logically omniscient reasoners. When defining situational awareness, one must consider the concepts of attention, vigilance, intelligence and stress within the context of resource-bounded agents.

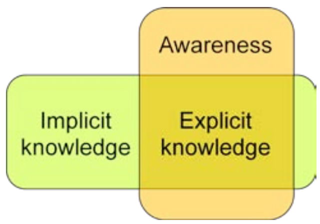

Figure 3: Awareness, implicit and explicit knowledge [10].

Therefore, we adopt the following definition of awareness: "an agent is aware of a proposition y if it can compute the truth value of y given its resources". The vulnerability awareness of one robot r_i [1] is thus directly linked to its ability to come with an answer the question y ="*Am I vulnerable*?" by means of an algorithm (*i.e.*, a classifier, as detailed in Section 4.2) given the limited resources available to the agent using this algorithm (power, memory, computation, move, etc). Particularly challenging is the feature selection process knowing that (1) the more features being used, the higher the computation and memory costs will be; and that (2) some feature may require a complete map of the swarm involving higher memory needs while other may be evaluated locally.

Features	**Space**	**Time**	**Required information**
Vertex betweenness	O(M)	O(M N)	Global
Degree	O(n)	O(n)	Local
Eigenvector centrality	O(M)	O(M)	Global
Kleinberg's authority score	O(M)	O(M)	Global
Average shortest path per node	O(M^2)	O(M N log$N$+M)	Global

n: the number of vertices to calculate
M: the number of vertices in the graph
N: the number of edges in the graph

Table 2: Some structural features and their computational complexity [4].

[1] This is a local definition but a global definition would concern a central instance having access to the global swarm's state.

Table 2 lists some features together with their computational complexity. They can be qualified as *local* or *global* if their computation requires the knowledge of the node information alone (*i.e.*, number of edges of this node) or the knowledge of the whole network. Clearly, a global feature requires more memory, and a complex feature requires more computation time to be extracted. Given their limited resources, the robots may not be aware of their vulnerability in the case that the computation time or memory required is higher than the allowed resources. The challenge is then to select the optimal subset of features together with the classification algorithm so that the robots are aware at any time of their vulnerability and possibly reorganize for improved robustness.

5.0 EXPERIMENTS

5.1 System description

5.1.1 Hardware description

The authors have access to real world robots that will allow them to implement the proposed design. The platform used is built on the iRobot Create [26] with a VIA Artigo Pico-ITX [27] computer mounted on top. The Pico processor is connected to the robot platform with a serial cable. The computer is capable of sending commands to and receives sensor readings from the robot platform by using the iRobot Create's Open Interface (OI). In the cargo bay of the robot, a battery is used to provide power for the computer.

5.1.2 Sensor description

The sensors on the iRobot Create provide us with odometry and collision detection. We use the USB ports on the Artigo computer to provide additional sensors which include a Logitech QuickCam Orbit AF camera [28], a Hokuyo URG-LX04 laser range finder [29], and an Alfa Network 802.11 b/g/n Wireless USB adapter [30]. The camera is used to provide vision but due to our limited processing power it is primarily used for blob detection. The laser range finder can be used for mapping, localization, and obstacle avoidance. The wireless adapter not only provides a means of communication but also allows us to determine the strength and quality of the signal coming from the other robots in the swarm.

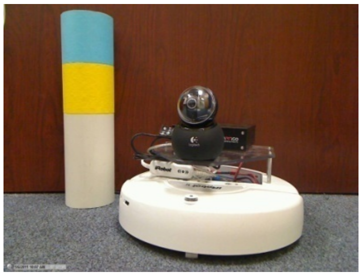

Figure 4: Robot node implemented with an iRobot Create.

5.1.2 Operating system description

The Artigo computer runs a Linux based operating system and each robot creates an ad-hoc connection to all other robots that are in range. We are also using the Robotics Operating System (ROS) [31] for hardware abstraction, device drivers, and to provide support for a publish-subscribe architecture. An added benefit of using ROS is that it provides us with the capability of quickly adding abilities like path planning or self-localization and mapping (SLAM).

5.2 Simulation results

In this section, we provide some results obtained through simulations. We randomly generated a series of 100 swarms of robots designed to cover a set of clients which positions are also randomly generated. Figure 5 shows such a network.

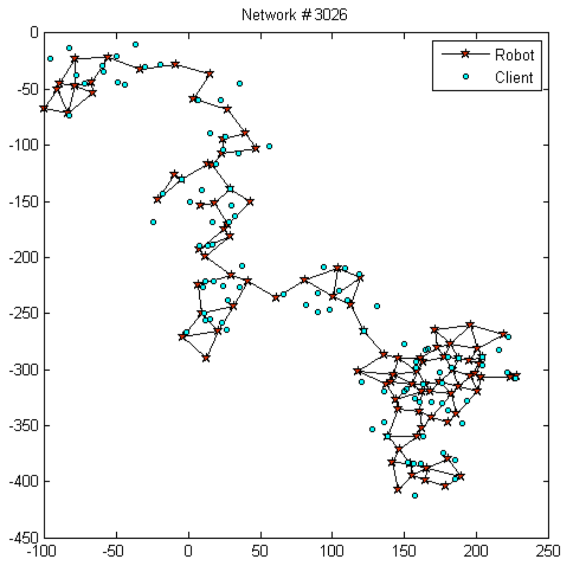

Figure 5: A tactical mobile cloud of 100 robots for communication coverage of 100 clients. Robots are represented by red stars, while clients are blue circles. The communication connection between two robots is represented by black lines.

First, each node of each network is labeled as vulnerable or non-vulnerable. For these experiments, a node is vulnerable if its removal breaks the network connectivity. Then, features are extracted for each node, leading to a table of features as described in Table 2. Besides the 11 structural features, a coverage feature is computed which is the number of clients covered by each robot.

A classifier is then trained to recognize vulnerable and thus predict vulnerabilities. We obtained a 69,7% recognition rate with a nearest mean classifier over a cross-validation error estimation with 5 folds and 10 repetitions. These preliminary results show that we are able to predict vulnerabilties within tactical swarms.

6.0 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a pattern recognition approach to network vulnerability assessment applied to a tactical swarm of robots to enhance their strategy for surveillance coverage. Each node of the tactical swarm is assigned a classifier which allows it to self-evaluate its vulnerability within the network based on structural features. Its awareness is linked to its limited resources in memory, computation time and power. We obtained encouraging results on the classification rate. The implementation proposed in this paper is a proof of concept that could be used on current and future systems by creating a swarm from a heterogeneous group of robots. This group could consist of a mixture of airborne robots such as unmanned airships like the Long Endurance Multi-Intelligence Vehicle (LEMV) [16] combined with ground robots such as the Small Unmanned Ground Vehicle (SUGV) [17] all acting autonomously to maintain network connectivity.

7.0 ACKNOWLEDGMENTS

This paper is based upon work supported by the U.S. Army Research Office under Grant Award Number MIPR9FDATXR048. The views expressed in this report are those of the authors and do not reflect the official policy or position of the United States Military Academy, the Department of the Army, or the Department of Defense.

8.0 REFERENCES

[1] Al Mannay, W. I. and Lewis, T. G., *Minimizing network risk with application to critical structure protection*, Journal of Information Warfare, Vol. 6, No. 2, pp. 52-68, 2007.

[2] Bezzo, N and Fierro, R., *Swarming of Mobile Router Networks*, 2011 American Control Conference, San Francisco, CA, 2011

[3] Csárdi, G., Nepusz, T., *The igraph software package for complex network research, Inter. Journal Complex Systems*, 1695, 2006.

[4] Godsile, C. and Royle, G., Algebraic Graph Theory, New York: Springer, 2001.

[5] Haimes, Y. Y., *On the definition of vulnerabilities in measuring risk to infrastructure*, Risk Analysis, Vol. 26, No. 2, pp. 293-296, 2006.

[6] Halpern, J., Moses, Y. and Vardi, M. Y. *Algorithmic knowledge*, in Proc. of the 5th Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'94). Morgan Kaufmann, 1994, pp. 255–266.

[7] Jousselme, A.-L., Maupin, P., Garion, G., Cholvy, L., Saurel, C., *Situation awareness and ability in coalitions*, 10th International Conference on Information Fusion, Quebec city, Canada, 9-12 July 2007.

[8] Léchevin, N., Rabbath, C. A., and Maupin, P., *Toward a stability monitoring system of an asset-communications network exposed to malicious attacks*, American Control Conf., San Francisco, 2011.

[9] Léchevin, N., Jousselme, A.-L., Maupin, P., *Pattern Recognition Framework for the Prediction of Network Vulnerabilities*, IEEE Network Science Workshop, West Point, NY, June 2011.

[10] Lee, J. and Ahn, C., *Improving Energy Efficiency Based on Behavioral Model in a Swarm of Cooperative Foraging Robots*, Genetic and Evolutionary Conference, Dublin (Ireland), July 2011.

[11] Levine, C., Soldiers as Gathers, *Analysers, and Users of Situational Information*, Workshop on Information Sharing at the Front Line, Indian Wells, CA, April 2010.

[12] McGill, W. L. and Ayyub, B. M., The meaning of vulnerability in the context of critical infrastructure protection, in Critical Infrastructure Protection: Element of Risk, Critical Infrastructure Protection Program, George Mason University School of Law, 2007.

[13] http://store.irobot.com/product/index.jsp?productId=2586252

[14] http://www.via.com.tw/en/products/embedded/artigo/a1000/index.jsp

[15] http://www.logitech.com/en-us/38/3480

[16] http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html

[17] http://www.alfa.com.tw/in/front/bin/ptdetail.phtml?Part=AWUS036NH

[18] http://www.ros.org

[19] http://www.as.northropgrumman.com/products/lemv

[20] http://www.bctmod.army.mil/systems/sugv/index.html